THE LINUX FOUNDATION PROJECTS

CIVIL INFRASTRUCTURE PLATFORM

Generating immutable, A/B updatable, securely booting Debian images

Jan Kiszka, Siemens AG

FOSDEM 2025, February 2nd 2025

About Us, About Me …

Civil Infrastructure Platform (CIP)

<cip-dev@lists.cip-project.org>

- Linux Foundation Project
- Enhance Linux for long-living industrial and critical infrastructure use cases
- SLTS kernel, real-time, testing, security certification, device updates, ...
- Upstream first, own projects second
- Members are suppliers and users in this domain

Jan Kiszka

<jan.kiszka@siemens.com>

- Siemens Foundational Technology
- (In-house) Embedded Linux consultant & developer
- CIP kernel workgroup chair, `isar-cip-core` maintainer
- Maintainer and contributor to various OSS projects

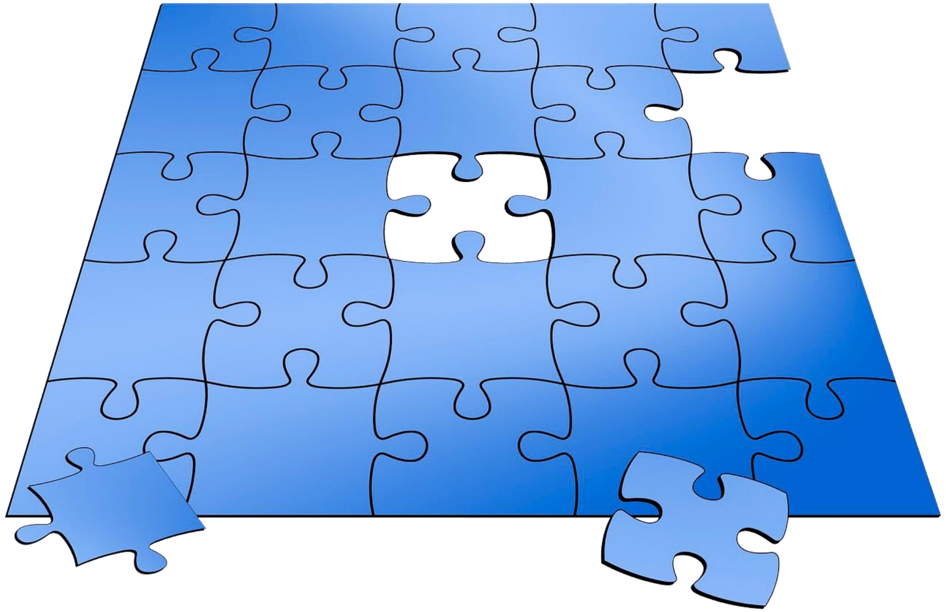
Why Debian? For Industrial Use Cases?

- Mature, high-quality, mainstream Linux distribution
- Support for many new and old hardware architectures
- Suitable for small and big installations
- Security updates, long-term support
- Strong OSS community

=> Selected as baseline for CIP

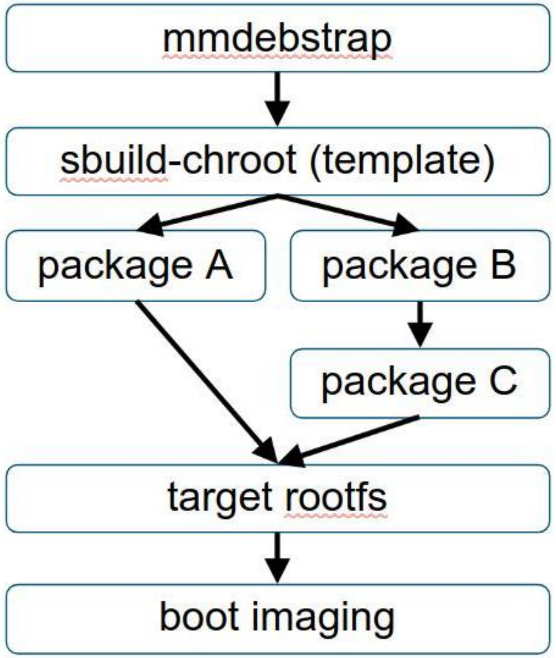
Missing Pieces in Our Puzzle

- Create flashable images for devices
- Never brick a device in the field!
  => Conservative A/B updates
- Create image update artifacts
- Enable secure boot
  - No gap between kernel and filesystem
  - With own keys typically
- Have a way customize few(!) packages



Isar [1] - Package and Image Build System for Debian

- Unique combination of
  - Package builder
  - Image creator
- Reuses Open Embedded bits
  - bitbake task engine
  - wic imager and plugins
  - OE libs for patching, caching etc.
- Recipes can be structured in layers
  - isar base => **isar-cip-core** => your project
- Using kas [2] for configuration management



mmdebstrap

sbuild-chroot (template)

package A

package B

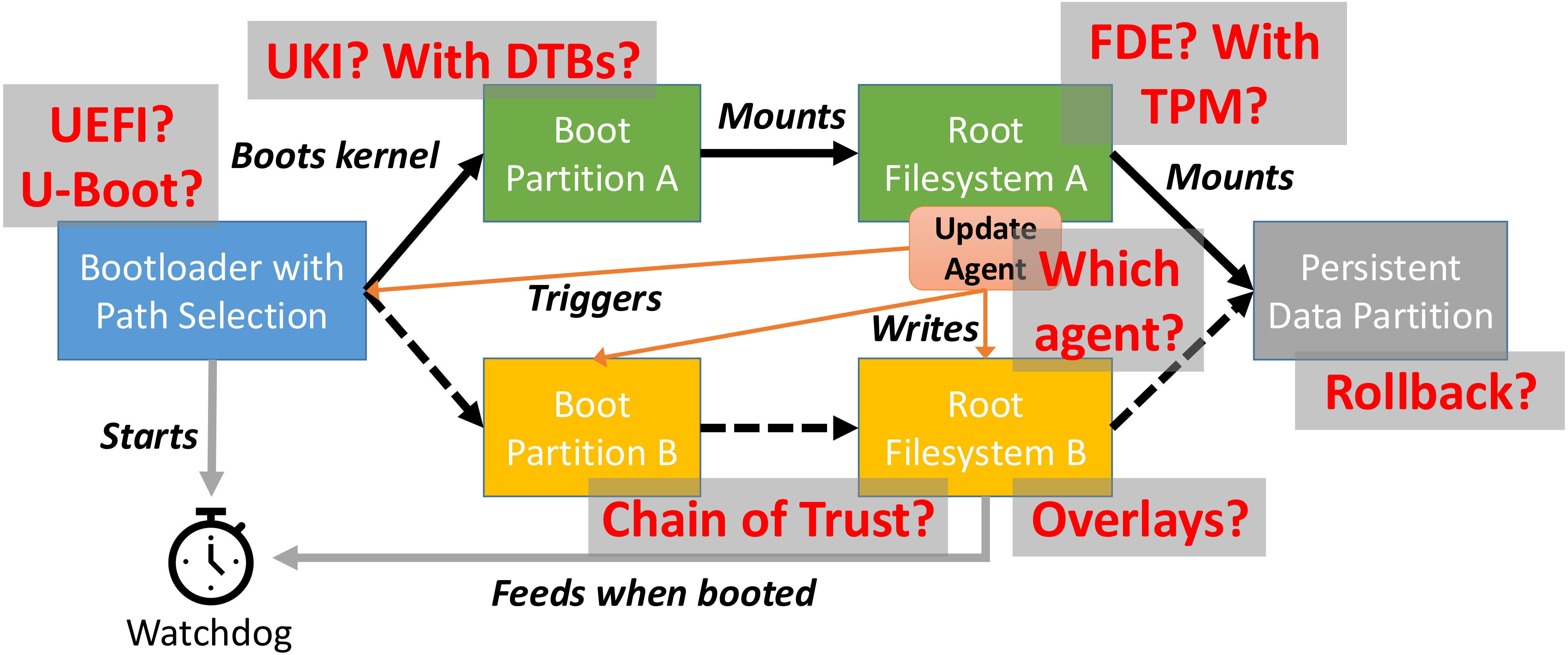
package C

target rootfs

boot imaging

[1] https://github.com/ilbers/isar
[2] https://github.com/siemens/kas

Dual-Copy (A/B) Update Pattern – Simple, No?



What isar-cip-core Provides

- UEFI-based boot pattern
  - EFI Boot Guard [1] as switcher and (x86) watchdog driver
  - Signed UKI images
- Key logic in initramfs hooks
  - A/B boot path chaining
  - Integrity for rootfs
  - Configurable overlays for read-only rootfs
  - Partition encryption with key in TPM
  - A/B snapshots/rollback for persistency [WIP]
- SWUpdate [2] as device update agent
  - Round-robin handler for A/B slots
  - Signed and optionally encrypted update artifacts
  - Delta image update support [3]
- Plumbing for r/o rootfs

[1] https://github.com/siemens/efibootguard
[2] https://github.com/sbabic/swupdate
[3] https://elinux.org/images/7/74/2024_EOSS_CIP_delta_updates.pdf

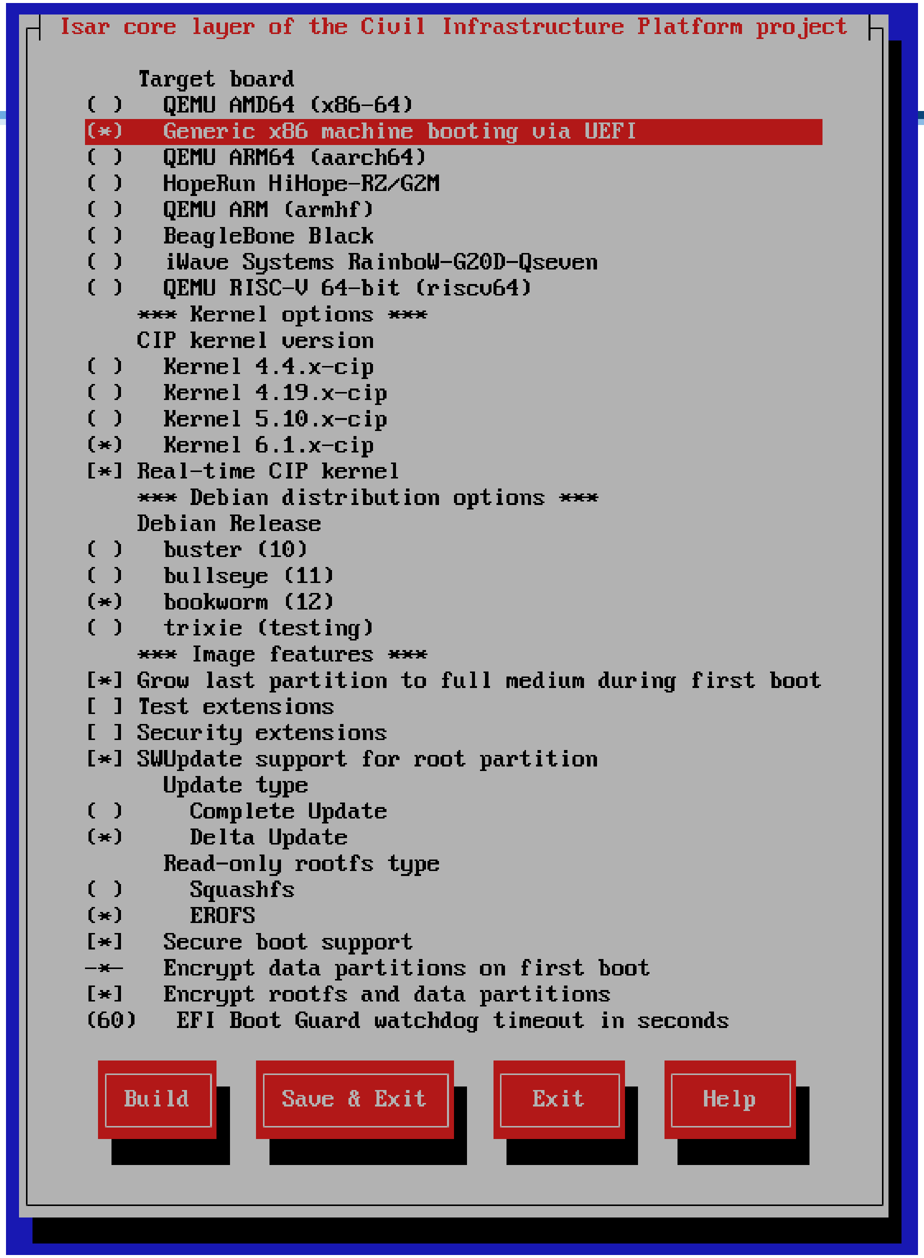
How Things Plug Together

- Off-device configuring, building and signing of initramfs / UKI
  - Signing helpers come with isar-cip-core
- Boot chain [1]
  - EFI Boot Guard selects UKI A or B
  - Image UUID and dm-verity hash included in initramfs, protected by UKI signature
  - Encrypted partitions unlocked with TPM (systemd or clevis)
  - dm-verity hash selects corresponding rootfs (squashfs, erofs)
  - Image UUID selects persistency snapshot (/var) [WIP]
  - Overlays mounted from /var as storage
- isar-cip-core images are for reference and testing
- Build your own project / product layer on top

[1] https://elinux.org/images/4/42/ELCE2022-UEFISecureBootOTAUpdatesOnARM.pdf

Want To Try It Out?

- Clone isar-cip-core
  https://gitlab.com/cip-project/cip-core/isar-cip-core
- Enable privileged docker or podman
- `./kas-container menu`
  - Supports x86, armhf, arm64 and riscv64
  - Full features only with latest releases
- `./start-qemu.sh`
  (ssh on localhost:22222)
- ...or flash to real device



```
 Isar core layer of the Civil Infrastructure Platform project

        Target board
    ( )   QEMU AMD64 (x86-64)
    (*)   Generic x86 machine booting via UEFI
    ( )   QEMU ARM64 (aarch64)
    ( )   HopeRun HiHope-RZ/G2M
    ( )   QEMU ARM (armhf)
    ( )   BeagleBone Black
    ( )   iWave Systems RainboW-G20D-Qseven
    ( )   QEMU RISC-V 64-bit (riscv64)
        *** Kernel options ***
        CIP kernel version
    ( )   Kernel 4.4.x-cip
    ( )   Kernel 4.19.x-cip
    ( )   Kernel 5.10.x-cip
    (*)   Kernel 6.1.x-cip
    [*] Real-time CIP kernel
        *** Debian distribution options ***
        Debian Release
    ( )   buster (10)
    ( )   bullseye (11)
    (*)   bookworm (12)
    ( )   trixie (testing)
        *** Image features ***
    [*] Grow last partition to full medium during first boot
    [ ] Test extensions
    [ ] Security extensions
    [*] SWUpdate support for root partition
          Update type
    ( )     Complete Update
    (*)     Delta Update
          Read-only rootfs type
    ( )     Squashfs
    (*)     EROFS
    [*]   Secure boot support
    -*-   Encrypt data partitions on first boot
    [*]   Encrypt rootfs and data partitions
    (60)   EFI Boot Guard watchdog timeout in seconds
```

Reproducible Images

- Reproducibility essential for supply chain security – and smaller delta updates
- Many Debian packages already reproducible
- CIP is supporting Reproducible Builds to close remaining gaps
- Many isar-cip-core images now reproducible
  - Tuned filesystem and update containers
  - Patched dosfstools (#1087568)
  - Worked with diffoscope to scan disk images
  - Weekly pipeline checks reproducibility

Reproducible Builds

Working with Debian Upstream

- Packaged of EFI Boot Guard & dependencies, took over maintenance
- SWUpdate
  - Worked with upstream to enable distro packaging (build-time -> runtime configuration, plugins)
  - Fixes and improvements of official package
- Worked with snapshot.debian.org on performance and stability improvements
- Still trying to avoid initramfs rebuilds (#1079509)
- More to come...

debian

Ongoing Work and Plans for the Future

- Finalize A/B snapshot of persistency
  - Filesystem recovery / reset
  - Review encryption approach for btrfs
  - Exclude problematic bits in /var (logs, containers, databases, …)
  - Provide alternatives (dm-snapshot, OSTree, …)
- Delta update for UKI
- Improve documentation
  - Many recipe APIs lack descriptions
  - Provide "hello world" skeleton layer
- Measured boot, possibly remote attestation
- Officially package initramfs bits for Debian ("iot-initramfs-tools", dracut module)?
- Explore & integrate alternative patterns

Summary

- Robust unattended software update, locked and secured – all possible with Debian, it "just" takes some plumbing
- CIP strives to provide reusable building blocks for this
  - Blueprints / pre-integrations
  - Testing and long-term maintenance
- Bits can be found at https://gitlab.com/cip-project/cip-core/isar-cip-core
- Join us at cip-dev@lists.cip-project.org

Thank You!

Questions?

CIVIL
INFRASTRUCTURE
PLATFORM